PATENT ABSTRACTS OF JAPAN

(11)Publication number : **10-003389**

(43)Date of publication of application : **06.01.1998**

(51)Int.CI. **G06F 9/38**

G06F 12/08

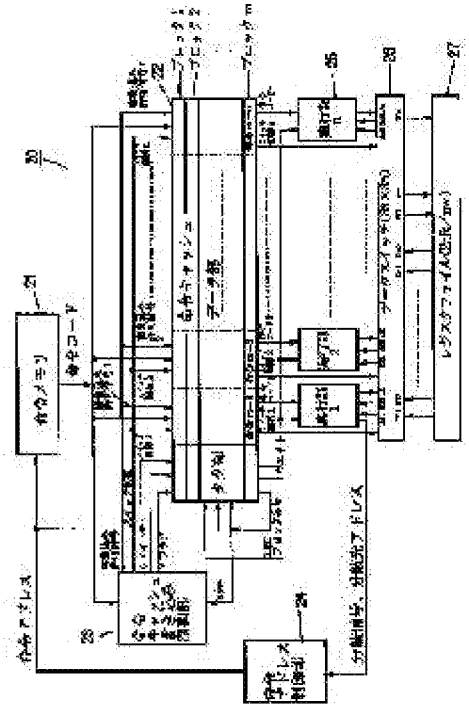
(21)Application number : **08-154069**

(22)Date of filing : **14.06.1996**

(71)Applicant : **OKI ELECTRIC IND CO LTD**

(72)Inventor : **KAWAI ATSUSHI**

(54) PARALLEL COMPUTER



(57)Abstract:

PROBLEM TO BE SOLVED: To provide a parallel computer with which circuit scale and the quantity of wiring are remarkably reduced, the reduction of chip size and the miniaturization of circuit delay are performed when making a CPU into LSI, an operating clock frequency is improved, power consumption is minimized and chip cost is reduced.

SOLUTION: A very long instruction word(VLIW) type parallel computer 20 is provided with an instruction memory 21, an instruction cache 22, an instruction cache write control part 23, an instruction address control part 24, plural instruction execution parts 25, a data switch 26 and a register file 27, and the instruction cache 22 is installed between the instruction memory 21 which is installed outside the CPU, and the instruction execution parts 25. Thus, a VLIW instruction is generated by continuously and successively reading plural VLIW element instructions from the instruction memory 22 into the instruction cache 22, and required wait information and information concerning the setting of path for bypassing executed result data are previously extracted by inspecting the dependency of register reference to the VLIW instruction generated just before.

LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

* NOTICES *
Japan Patent Office is not responsible for any damages caused by the use of this translation.
1. This document has been translated by computer.So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the parallel computer which performs simultaneously processing which became independent, respectively in two or more instruction-execution sections Instruction memory, an instruction cache, an instruction-cache write-in control section, an instruction address control section, It has two or more instruction-execution sections, a data switch, and a register file. next, when the VLIW (Very Long Instruction Word) instruction which should be executed does not exist in the aforementioned instruction cache Every one instruction code used as the element of a VLIW instruction is continuously read from the aforementioned instruction memory for every clock cycle. It sets to the aforementioned instruction-cache write-in control section at the same time it writes in the block made into the write-in object in the aforementioned instruction cache serially. Consist of aforementioned instruction code and instruction code of plurality respectively which read into just before from the aforementioned instruction memory, and was generated. When the register reference dependence between one or more VLIW instruction codes is inspected and reference dependence is not detected The value as

which it was beforehand determined for supplying the read-out data from the aforementioned register file to the aforementioned instruction-execution section as data switch control information When it writes in the applicable block of the aforementioned instruction cache and reference dependence is detected The number of wait cycles for inhibiting execution of the next VLIW instruction is extracted. The aforementioned data switch is minded for the execution result data of the VLIW instruction executed immediately before from the aforementioned instruction-execution section without the aforementioned register file at the same time it writes this number of wait cycles in the applicable block of the aforementioned instruction cache. In order to supply the specific instruction-execution section of two or more aforementioned instruction-execution sections directly, Extract data switch control information and this data switch control information is written in the applicable block of the aforementioned instruction cache. Then, the newly generated applicable VLIW instruction is read in the aforementioned instruction cache. The read VLIW instruction All the instruction-execution sections, a data switch, and register stuffer IRUHE, When the VLIW instruction which should supply simultaneously, should execute this VLIW instruction and should next be executed exists in the aforementioned instruction cache The parallel computer which reads the VLIW instruction which should be this executed from the applicable block of the aforementioned instruction cache, and is characterized by all the instruction-execution sections, the data switch and register stuffer IRUHE, and constituting so that it may supply simultaneously and this VLIW instruction may be executed.

[Claim 2] In the parallel computer which performs simultaneously processing which became independent, respectively in two or more instruction-execution sections Instruction memory, an instruction cache, an instruction-cache write-in control section, an instruction address control section, Have two or more instruction-execution sections, a data switch, and a register file, and from the aforementioned instruction memory, read 1 instruction code for every clock cycle, and it sets to the aforementioned instruction-cache write-in control section. it read immediately before -- it consists of instruction codes of n (n is arbitrary integers) individual, respectively -- When register reference dependence with one or more VLIW instruction codes is inspected and reference dependence is detected While generating the number of wait cycles for inhibiting execution of the next VLIW instruction The switch control information for performing data forwarding is generated. the aforementioned number of wait cycles, and the aforementioned switch control information It writes in the applicable block of an instruction cache simultaneously with the read instruction code, and the VLIW instruction is beforehand constituted in the aforementioned instruction cache. at the time of a VLIW instruction execution The VLIW instruction code which does not perform the register reference dependence check during a VLIW instruction, but corresponds from the aforementioned instruction cache, The parallel computer characterized by constituting so that data forwarding by the change of the VLIW instruction-execution queuing and the aforementioned data switch for referring to the VLIW instruction-execution result which read simultaneously the number of weight and switch control information, and was performed immediately before may be performed.

[Claim 3] The execution time of the aforementioned instruction is a parallel computer given in the claim 1 or any of 2 they are. [which is characterized by being one clock cycle]

[Claim 4] The execution time of the aforementioned instruction is a parallel computer given in the claim 1 or any of 2 they are. [which is characterized by being two or more clock cycles]
[Claim 5] The aforementioned parallel computer is a parallel computer given in the claim 1 or any of 2 they are. [which is characterized by being a VLIW (Very Long Instruction Word:super-long-form-of-length machine instruction) type parallel computer]

[Translation done.]

* NOTICES *
Japan Patent Office is not responsible for any damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[The technical field to which invention belongs] this invention relates to a parallel computer and relates to improvement of the instruction-execution control section in a VLIW type parallel computer (Very Long Instruction Word:super-long-form-of-length machine instruction type parallel computer) especially.
[0002]
[Description of the Prior Art] There is a method of publishing two or more instructions per one clock cycle. This method enables an instruction execution rate to exceed a clock rate.
[0003] Since a compiler performs a VLIW type parallel computer (Very Long Instruction Word:super-long-form-of-length machine instruction type parallel computer) with responsibility till the place which summarizes truly the instruction that can be published simultaneous and made into one super-long-form-of-length machine instruction, hardware's does not need to waver at all about the concurrency possibility of an instruction.
[0004] For example, there are some which were indicated by "quantitive approach of a HENESHI & Paterson computer architecture -design, realization, and evaluation -" (Nikkei Business Publications, 1992, pp 312-323) as this kind of a parallel computer.
[0005] Drawing 9 is drawing showing the composition of the conventional VLIW type parallel computer, and a VLIW type parallel computer consists of instruction memory 11, a statement part 12 (statement-part 1-n), a data switch 13, and a register file 14 in this drawing.
[0006] The above-mentioned instruction memory 11 supplies an instruction to ********** to a statement part 1 - n, respectively. For this reason, the

instruction length read from instruction memory 11 at once is set to instruction-length xn given to each statement part. The instruction length given to each statement part is about 32 bits like the usual common computer.
[0007] Therefore, the instruction length read from instruction memory 11 becomes 32xn bits, and becomes very long. This is the reason called VLIW type parallel computer.
[0008] Each statement part executes the instruction supplied from instruction memory 11 for every instruction execution cycle. On the other hand, operand data is outputted to an input or a register file 14 from a register file 14 through a data switch 13.
[0009] In order for all statement parts to be surely able to write the arbitrary registers in a register file 14 in reference or a register, each statement part calculates to two input operands, and considering writing one output operand in a register, the multi-port register file of a 2xn input and n output is needed.
[0010] Similarly, a data switch 13 turns into a 3nx3n (3n input, 3n output) crossbar switch, in order not to blockade.
[0011] Furthermore, in order [of n next instructions of n last instruction-execution result data] to make reference respectively possible as two operand data, there are many things equipped with the mechanism which investigates the register reference dependency during the VLIW instruction with n instruction codes each, changes a data switch if needed, and supplies again not the read-out data of a register file 14 but the output data from a statement part to a statement part.
[0012]
[Problem(s) to be Solved by the Invention] In such a conventional VLIW type parallel computer, since it is necessary to each statement part to give a separate instruction to **********, VLIW instruction code (hereafter, the instruction code in a VLIW computer performed at once is called a VLIW instruction or VLIW instruction code, and each instruction code performed by each statement part is only called an instruction or instruction code.) becomes huge. For this reason, the buffer with the instruction memory which installed the instruction cache in the interior of CPU, and was installed in the CPU exterior, and the instruction-execution section is performed.
[0013] Though instruction memory is made into the read-out width of face for one instruction like the usual computer by this, there are many whose reference pack from an instruction cache to one VLIW instruction, and is enabled. In such a VLIW computer, in order to minimize the queuing time of the instruction execution by the check and reference dependence of data reference dependence during each VLIW instruction, there are many things equipped with a data forwarding mechanism (data bypass mechanism for referring to the instruction-execution result data from a statement part without a register file for the following instruction execution).
[0014] However, the circuit scale and the amount of wiring for the combination of the reference dependence during a VLIW instruction checking very many (order of N2) this become great. For this reason, in case CPU is LSI-ized, problems, such as increase of a chip size and a fall of the clock frequency of operation by circuit delay, are caused.
[0015] It aims at offering the parallel computer which can reduce chip cost while it becomes possible [minimizing reduction of a chip size, and circuit delay], raises a clock frequency of operation and minimizes power consumption, in case this invention can cut down a circuit scale and the amount of wiring sharply and LSI-izes CPU.
[0016]

[Means for Solving the Problem] In the parallel computer to which the parallel computer concerning this invention performs simultaneously processing which it became independent of, respectively in two or more instruction-execution sections Instruction memory, an instruction cache, an instruction-cache write-in control section, an instruction address control section, It has two or more instruction-execution sections, a data switch, and a register file. next, when the VLIW (Very Long Instruction Word) instruction which should be executed does not exist in an instruction cache Every one instruction code used as the element of a VLIW instruction is continuously read from instruction memory for every clock cycle. It sets to an instruction-cache write-in control section at the same time it writes in the block made into the write-in object in an instruction cache serially. Consist of instruction code and instruction code of plurality respectively which read into just before from instruction memory, and was generated. When the register reference dependence between one or more VLIW instruction codes is inspected and reference dependence is not detected The value as which it was beforehand determined for supplying the read-out data from a register file to the instruction-execution section as data switch control information When it writes in the applicable block of an instruction cache and reference dependence is detected The number of wait cycles for inhibiting execution of the next VLIW instruction is extracted. A data switch is minded for the execution result data of the VLIW instruction executed immediately before from the instruction-execution section without a register file at the same time it writes this number of wait cycles in the applicable block of an instruction cache. The data switch control information for supplying the specific instruction-execution section of two or more instruction-execution sections is extracted directly. This data switch control information is written in the applicable block of an instruction cache. Then, the newly generated applicable VLIW instruction is read in an instruction cache. The read VLIW instruction All the instruction-execution sections, a data switch, and register stuffer IRUHE, When the VLIW instruction which should supply simultaneously, should execute this VLIW instruction and should next be executed exists in an instruction cache The VLIW instruction which should be this executed is read from the applicable block of an instruction cache, and all the instruction-execution sections, a data switch and register stuffer IRUHE, and **** are supplied, and it constitutes so that this VLIW instruction may be executed.

[0017] Moreover, the parallel computer concerning this invention is simultaneously set in two or more instruction-execution sections. In the parallel computer which performs processing which became independent, respectively Instruction memory, an instruction cache, An instruction-cache write-in control section, an instruction address control section, two or more instruction-execution sections, Have a data switch and a register file, and from instruction memory, read 1 instruction code for every clock cycle, and it sets to an instruction-cache write-in control section. it read immediately before -- it consists of instruction codes of n (n is arbitrary integers) individual, respectively -- When register reference dependence with one or more VLIW instruction codes is inspected and reference dependence is detected While generating the number of wait cycles for inhibiting execution of the next VLIW instruction The switch control information for performing data forwarding is generated. the number of wait cycles, and switch control information It writes in the applicable block of an instruction cache simultaneously with the read instruction code, and the VLIW instruction is beforehand constituted in the

instruction cache. at the time of a VLIW instruction execution The VLIW instruction code which does not perform the register reference dependence check during a VLIW instruction, but corresponds from an instruction cache, The number of weight and switch control information are read simultaneously, and it constitutes so that VLIW instruction-execution queuing for referring to the VLIW instruction-execution result performed immediately before and data forwarding by the change of a data switch may be performed.

[0018] Moreover, the execution time of an instruction may be one clock cycle, and the execution times of an instruction may be two or more clock cycles.

[0019] Moreover, a parallel computer may be a VLIW (Very Long Instruction Word:super-long-form-of-length machine instruction) type parallel computer.

[0020]

[Embodiments of the Invention] The parallel computer concerning this invention is applicable to a VLIW type parallel computer.

[0021] Drawing 1 is drawing showing the whole VLIW type parallel computer composition concerning the 1st operation gestalt of this invention.

[0022] In drawing 1 , the VLIW type parallel computer 20 consists of instruction memory 21, an instruction cache 22, the instruction-cache write-in control section 23, the instruction address control section 24, a statement part 25 (statement-part 1-n), a data switch 26, and a register file 27.

[0023] According to the instruction address inputted from the instruction address control section 24, the above-mentioned instruction memory 21 reads one instruction code (instruction code performed by one statement part) for every clock cycle, and gives it to an instruction cache 22 and the instruction buffer write-in control section 23. An instruction cache 22 is equipped with m blocks which consist of the tag section and data division.

[0024] Drawing 2 is drawing showing the composition of the above-mentioned instruction cache 22.

[0025] In drawing 2 , as for an instruction cache 22, the tag section consists of a V flag, the instruction address, and a LRU flag.

[0026] V in drawing 2 means a Valid flag, and shows whether they are whether the content of an applicable block is effective, and no. The number of instructions is counted from instruction 1 among each instruction stored in instruction 1 - n, and shows the effective number of instructions. A number of instructions shown with this number of instructions show that parallel execution is possible in a statement part 1 - n.

[0027] The instruction address shown in the hatching portion of drawing 2 shows the instruction address arranged on the instruction memory of the instruction code stored in the instruction 1 of an applicable block. When the value which is in agreement with the instruction address given from the outside is stored in this instruction address section, it is shown that the instruction code which should be referred to is stored in an applicable block.

[0028] LRU means a Least Recentyly Used flag, and when this flag is 0, it shows that the instruction stored in the applicable block is not referred to recently. Such a block serves as a candidate for updating at the time of storing a new instruction code train in an instruction buffer.

[0029] The above-mentioned instruction address section and the LRU section consist of

content-retrieval type memory (associative memory), and when in agreement with the content stored in the applicable portion, they have the structure which outputs a data requirement.
[0030] The above-mentioned instruction address section asserts a coincidence signal, when the block which is in agreement with the content of the instruction address inputted exists (logic "1"), and it outputs the content of V flag of an applicable block, the number of instructions and instruction 1 - n.
[0031] Moreover, the LRU section outputs the block number of the watch young No. 1 whose V flag is 0, when setting LRU of an applicable block to 1 when read-out is performed, and writing in. V flag of an applicable block is set to 0 at the same time LRU outputs the block number young No. 1 which is 0, when all V flags are 1. V flag of an applicable block is set to 0 at the same time it outputs the block of old age watch, i.e., m, most, when all V flags and all LRU are 1. Moreover, at the time of writing (at the time of renewal of a block), only the LRU flag of the updated block is set to 1, and the LRU flag of other the blocks of all is set to 0.
[0032] Furthermore, the data division of the above-mentioned instruction cache 22 consist of the n instruction storing sections, and each instruction storing section consists of a portion which stores instruction code, and a portion which stores switch control information.
[0033] The switch control information storing section consists of source operand 1 specification part and source operand 2 specification part, respectively.
[0034] When it returns to drawing 1 and the VLIW instruction which should next perform the above-mentioned instruction-cache write-in control section 23 does not exist in an instruction cache 22 (henceforth an instruction-cache mistake) Similarly, the case where it is not an instruction-cache mistake is called instruction-cache hit. While performing control for storing in an instruction cache 22 the instruction code read from instruction memory 21 one by one, the register reference dependence between one generated within the instruction cache 22 immediately before or two or more VLIW instructions is checked.
[0035] When reference dependence is detected, the switch control information (switch control) for carrying out the direct reference of the required data from a statement part through a data switch 26 without a register file 27 is outputted, and it has a mechanism for storing in the applicable block of an instruction cache 22. In addition, about the detailed circuitry of the instruction-cache write-in control section 23, it mentions later by drawing 4 .
[0036] With this operation gestalt, all instruction executions shall be completed by one clock cycle in each statement part.
[0037] Drawing 3 is drawing showing the format of the instruction code performed by the above-mentioned VLIW type parallel computer 20.
[0038] In drawing 3 , there are two sorts, operation instruction and branch instruction, in an instruction, and it has a respectively separate format. Operation instruction consists of operation specification, source 1 operand, source 2 operand, and a destination operand. Moreover, branch instruction consists of branching specification and the branching place address.
[0039] The operation specified by operation specification performs to the content of the register specified with the content and the source operand 2 of the register specified with

the source operand 1 in operation instruction, operation store a result in the register specified with a destination operand performs, and when satisfied [with branch instruction] of branching specification conditions, the operation which makes into the instruction address which should next perform the branching place address carries out.
[0040] Drawing 4 is the circuit diagram showing the composition of the instruction-cache write-in control section 23 in the above-mentioned VLIW type parallel computer 20.
[0041] In drawing 4 , the instruction-cache write-in control section 23 consists of an instruction register 31, a shift register 32 (shift register 1-n), latch 33 (latches 11-1n), a comparator 34 (comparator 11-n2), a buffer 35 (buffer 11-n2), an instruction counter 36, a decoder 37, AND circuit 38 (AND circuit 1), AND circuit 39 (AND circuit 2), OR circuit 40 (OR circuit 1), and D latch 41.
[0042] The above-mentioned instruction register 31 is a register for holding temporarily the instruction code read from instruction memory 21.
[0043] The above-mentioned shift register 32 (shift register 1-n) saves the content of each destination operand of n instruction codes continuously read from instruction memory 21 one by one, and it holds the content until n instruction codes are stored in an instruction cache 22 and generation of a VLIW instruction is completed.
[0044] The above-mentioned latch 33 (latch 1-n) is for holding the destination operand of each instruction code which constitutes the VLIW instruction generated immediately before, respectively, and it holds the content until a new VLIW instruction is generated in an instruction cache 22.
[0045] The source operand 1 which refers to the above-mentioned comparator 34 (comparator 11-n2) by the instruction code newly held at the instruction register 31, respectively, and the source operand 2 inspect whether it is in agreement with n destination operands specified by the VLIW instruction generated just before being stored in latch 1 - n, and the reference dependence during a VLIW instruction is checked.
[0046] The above-mentioned buffer 35 (buffer 11-n2) It is what outputs the destination operand of the VLIW instruction generated just before being held at latch 1 - n, respectively, when coincidence is detected by comparator 11-n2. An OR is taken, respectively, and as the data forward 1 and a data forward 2, buffer 11 and 21 - n1 output and a buffer 12, and 22-n2 are stored in the applicable block of an instruction cache 22, in order to perform switch control.
[0047] a ** [while the above-mentioned instruction counter 36 reads instruction code from instruction memory 21 at the time of an instruction-cache mistake, storing in an instruction cache 22 and generating a VLIW instruction] clock cycle -- counting is performed An instruction counter 36 carries out counting of the number of instruction codes for VLIW instruction composition, and is newly initialized by the carry output (set to 1). Moreover, a carry output asserts V flag (it is made 1).
[0048] It will be shown by this V flag being written in an instruction cache 22 that the content of the corresponding block is effective.
[0049] counting of an instruction counter 36 -- an output is decoded by the decoder 37 and serves as a write-in enabling signal to each instruction code of an instruction cache 22, and the write-in position of switch control information, respectively
[0050] Above-mentioned AND circuit 38 (AND circuit 1) is a gate circuit for supplying a clock to each part only at the time of an instruction-cache mistake.
[0051] The above-mentioned D latch 41, OR circuit 40 (OR circuit 1), AND circuit 39

(AND circuit 2), and a flip-flop 42 constitute the changes detector 43 by which only the first one clock cycle which the instruction-cache mistake generated as a whole is asserted.
[0052] The above-mentioned changes detector 43 serves as information for this output inhibiting one clock cycle of execution of the VLIW instruction newly generated as a weight signal, and is written in the applicable block of an instruction cache 22.
[0053] When an instruction cache 22 makes a mistake in this weight information first Since register reference dependence with the VLIW instruction read from the instruction cache 22 just before it (carrying out an instruction-cache hit) cannot be inspected Since data forwarding is not made, it is for performing the VLIW instruction which inserts a wait cycle compulsorily, reads from a register file even if it is as a result of [last] a VLIW instruction execution, and is executed next.
[0054] The instruction address control section 24 shown in drawing 1 generates the instruction address which should read to a degree and should be performed, and shows circuitry to drawing 5 .
[0055] Drawing 5 is the circuitry view of the above-mentioned instruction address control section 24, and the instruction address control section 24 consists of instruction address counters 51 in this drawing.
[0056] In the state (negation state) where the branching signal inputted is not asserted, the increment of the instruction address counter 51 is carried out every [n] for every clock cycle, and it generates the instruction address in the case of executing a VLIW instruction sequentially. Moreover, where a branching signal is asserted, the branch address inputted is set up as initial value of a counter, and the branching place instruction address is generated.
[0057] Returning to drawing 1 , the above-mentioned statement part 25 (statement-part 1-n) is a statement part of a VLIW instruction, and performs n processings simultaneously by one clock cycle. Since two or more branch instruction cannot be executed simultaneously, the VLIW instruction containing branch instruction makes branch instruction instruction code surely performed by the statement part 1, and other instruction codes (it performs by the statement part 2 - n) serve as NOP (Non OPeration : instruction which processes nothing).
[0058] The above-mentioned register file 27 stores the SOSUPE land data for an operation, or stores the destination operand data which is the result of an operation. Two source operand data are independently supplied to n statement parts simultaneous, respectively, and in order to make it possible to write in one destination data independently, respectively, it has 2n read-out Pau ** and n write-in ports.
[0059] The above-mentioned data switch 26 is a data switch for supplying 2n source operand data simultaneously, and supplying n destination data simultaneously to a register file 27 to a statement part 1 - n, and has composition which enables "to refer to the data between the statement parts for performing data forwarding." For this reason, it consists of crossbar switches of a 3n input x3n output.
[0060] Hereafter, operation of the VLIW type parallel computer 20 constituted as mentioned above is explained.
[0061] In the VLIW type parallel computer 20 shown in drawing 1 , read-out of instruction memory 21 and the index of an instruction cache 22 are simultaneously performed by the instruction address outputted from the instruction address control section 24.

[0062] The block holding the instruction address equal to the instruction address inputted exists in an instruction cache 22, and when V flag of the block is 1, it is recognized as an instruction-cache hit. In this case, required information is read from the applicable block of an instruction cache 22.
[0063] Required information is weight information and instruction 1-n, as shown in the composition of the instruction cache of aforementioned drawing 2 .
[0064] Although it is restricted when it is the VLIW instruction by which the VLIW instruction which next performs this was first generated by the instruction-cache mistake when weight information had the unknown register reference dependence between the VLIW instruction executed immediately before and the VLIW instruction executed to a degree After inhibiting one clock cycle of execution of the next VLIW instruction compulsorily and writing the last instruction-execution result in a register file 27, the synchronization for making read-out of the source operand data for the following VLIW instruction execution start is taken.
[0065] Instruction 1 - n consist of switch control information and instruction code, respectively.
[0066] Moreover, switch control information consists of switch-off substitute information required in order to give source operand 1 data of SW11-n1 to a statement part, and switch-off substitute information required in order to give source operand 2 data of SW12-n2 to a statement part.
[0067] Each switch control information of SW11-n1, and SW12-n2 is SW11, 12-SWn1, and the pair of n2, respectively, and is written in the applicable block of an instruction cache 22 as each switch control information of instruction 1 - n.
[0068] When recognized as switch control information not having the register reference dependence during a VLIW instruction in the instruction-cache write-in control section 23, it is not necessary to perform data forwarding. At this time, the input position to the data switch 26 of the read-out data from a register file 27 only serves as switch information.
[0069] That is, as SW11, like [R12] the following as SW12, Rn1 is written in SWn1 and Rn2 is written for R11 of a data switch in an instruction cache 22 as SWn2 in drawing 1 . When recognized as there being register reference dependence during a VLIW instruction, the switch control information 1, i.e., the data forward, and the data forward 2 for performing data forwarding supplied from the instruction-cache write-in control section 23 are written in the applicable block of an instruction cache 22.
[0070] All the switch control information of the applicable block read from an instruction cache 22 is simultaneously supplied to a data switch 26.
[0071] Moreover, instruction code 1 - n are supplied to a statement part 1 - n, respectively. All of each source operand 1, the source operand 2 (these specify the register used as source operand data), and destination operand (the register which stores destination operand data is specified) of instruction code 1 - n are simultaneously supplied also to a register file 27. In addition, this signal connection is omitted in drawing 1 .
[0072] In a register file 27, 2n source operand data specified with each source operand 1 contained in the instruction code 1 supplied from the instruction cache 22 - n and the source operand 2 is read simultaneously, and these all are supplied to a data switch 26.
[0073] According to the switch control information supplied from the instruction cache

22, a crossbar switch is controlled by the data switch 26. For example, when the source operand 1 of instruction code 1 specifies 27 register filer1 and there is no reference dependence about this register, data forwarding is not performed, but since the value of 27 register filer1 inputted into a data switch 26 and this are inputted into 26 data switchR11 shown in drawing 1 , a switch is changed so that R11 of drawing 1 may be connected to S11.

[0074] Moreover, when the source operand 2 of instruction code n specifies 27 register filer2 and this register is updated by execution of the instruction n of the last VLIW instructions, data forwarding is performed, and since the execution result of the instruction n inputted into a data switch 26 and this are inputted into Dn of the data switch 26 shown in drawing 1 , a switch is changed so that Dn of drawing 1 may be connected to Sn2.

[0075] Each source operand 1 data given to a statement part 1 - n and source operand 2 data are calculated in a statement part 1 - n through S11-S2n of a data switch 26, and the destination operand data which is result data is outputted to D1-n of a data switch.

[0076] In a data switch 26, while making it connect with W1-n, respectively and supplying the destination operand data inputted into D1-n, the destination operand data which had data forwarding specified in switch control information is connected to the predetermined output of the S11-2n as source operand data for the VLIW instruction executed next.

[0077] In a register file 27, the destination data inputted from W1-n of a data switch 26 are written in the register specified with the destination operand of instruction code 1 - n, respectively.

[0078] Since the operation top of a program is impossible also for executing two or more branch instruction simultaneously and executing other instructions simultaneously with branch instruction, ****** and others when branch instruction is contained in a VLIW instruction, in this case, as for all of instruction code 2 - n, branch instruction becomes instruction code 1 with NOP. Branch instruction is surely executed by the statement part 1. A branching signal is asserted, a branch address is outputted and a result is supplied to the instruction address control section 24.

[0079] In the instruction address control section 24 shown in drawing 5 , in the state (negation state) where the branching signal inputted is not asserted, the increment of the instruction address counter 51 is carried out every [n] for every clock cycle, and it generates the instruction address in the case of executing a VLIW instruction sequentially. Moreover, where a branching signal is asserted, the branch address inputted is set up as initial value of a counter, and the branching place instruction address is generated.

[0080] When the instruction address equal to the instruction address inputted does not exist in the tag section in the instruction-cache mistake 22, i.e., an instruction cache or even if it exists, when V flag of the block is 0 While newly reading n instruction codes from instruction memory 21 and writing these in the updating place block of an instruction cache 22 one by one When register reference dependence with the VLIW instruction generated by another block in an instruction cache 22 is checked and reference dependence is detected immediately before The switch control information for performing data forwarding is generated, this is written in the applicable block of an instruction cache 22, this is supplied to a data switch 26 at the time of a VLIW instruction

execution, and the path for a data switch 26 performing data forwarding is formed.
[0081] Thus, it is the instruction-cache write-in control section 23 to control writing to the generation of switch control information for the operand data supply at the time of the writing to an instruction cache 22 and VLIW instruction execution of the instruction code read from instruction memory 21 on the occasion of an instruction-cache mistake and the instruction cache 22 of this.
[0082] Hereafter, operation of the instruction-cache write-in control section 23 shown in drawing 4 is explained in detail.
[0083] In the instruction-cache write-in control section 23 shown in drawing 4 , the instruction code read from instruction memory 21 by the instruction address at the time of an instruction-cache mistake is first stored in an instruction register 31.
[0084] In a shift register 32 (shift register 1-n), the contents are held until it saves, n instruction codes are stored in an instruction cache and generation of a VLIW instruction is completed, carrying out shift in of the contents of each destination operand of n instruction codes continuously read from instruction memory 21 one by one, and carrying out a shift-out after 1 clock cycle.
[0085] Latch 33 (latch 1-n) is for holding the destination operand of each instruction code which constitutes the VLIW instruction generated immediately before, respectively, and it holds the contents until a new VLIW instruction is generated in an instruction cache 22.
[0086] That is, in latch 1-n, n destination operands of the VLIW instruction code generated in the instruction cache 22 immediately before are held until a VLIW instruction new next is generated within an instruction cache 22. Therefore, the register reference dependence during a VLIW instruction can be checked by comparing with the content of latch 1 - n the source operand 1 of the instruction code held at the instruction register 31, and the source operand 2, respectively. This comparison is performed in the comparator 34 (comparator 11-n2) of drawing 4 .
[0087] Moreover, the buffer 35 (buffer 11-n2) It is what outputs the destination operand of the VLIW instruction generated just before being held at the latch 33 (latch 1-n), respectively, when coincidence is detected by comparator 11-n2. An OR is taken, respectively, and as the data forward 1 and a data forward 2, buffer 11 and 21 - n1 output and a buffer 12, and 22-n2 are stored in the applicable block of an instruction cache 22, in order to perform switch control.
[0088] In above-mentioned comparator 11-n1 and above-mentioned comparator 12-n2, it is supposed that coincidence will be detected only in at most one comparator, respectively. In one VLIW instruction, two or more instructions of this are because it is carrying out storing each result of an operation in the same register if it cannot be simultaneously.
[0089] Therefore, the switch control information for performing a maximum of two data forwarding of the data forward 1 and the data forward 2 is generated to one instruction code.
[0090] moreover, a ** [while the instruction counter 36 of drawing 4 carries out counting of the number of instruction codes for VLIW instruction composition, reading instruction code from instruction memory 21 at the time of an instruction-cache mistake, storing in an instruction cache 22 and generating a VLIW instruction] clock cycle -- counting is performed It is newly initialized by the carry output (set to 1). Moreover, a carry output asserts V flag (it is made 1). The writing of the corresponding block will be

completed by this V flag being written in an instruction cache 22, and it will be shown that the contents are effective.
[0091] moreover, counting of an instruction counter 36 -- an output is decoded by the decoder 37 and serves as a write-in enabling signal to each instruction code of an instruction cache 22, and the write-in position of switch control information, respectively
[0092] AND circuit 38 (AND circuit 1) is a gate circuit for supplying a clock to each part only at the time of an instruction-cache mistake.
[0093] The D latch 41, OR circuit 40 (OR circuit 1), AND circuit 39 (AND circuit 2), and a flip-flop 42 are the changes detectors 43 which generate the signal with which the Arthur ** only of the first one clock cycle which the not continuous instruction-cache mistake generated is carried out.
[0094] An instruction-cache mistake is canceled by collating of the instruction address being in agreement and V flag being asserted by it, if a VLIW instruction is newly generated in an instruction cache 22 by instruction-cache mistake.
[0095] The VLIW instruction just generated by this is read from an instruction cache 22, and is supplied to a statement part 1 - n. However, in an instruction-cache mistake, a mistake signal is asserted from an instruction cache 22 after 1 clock cycle also about the VLIW instruction just behind this again. That is, as for a mistake signal, in an instruction-cache mistake, between one clock cycles is negated continuously.
[0096] Therefore, in the state where the mistake signal was negated two or more clock cycles, the time of the beginning of a not continuous instruction-cache mistake is detectable by detecting the changes by which the mistake signal was newly asserted.
[0097] As a weight signal, this signal generated by the changes detector 43 serves as information for inhibiting one clock cycle of execution of the VLIW instruction newly generated, and is written in the applicable block of an instruction cache 22.
[0098] When an instruction cache 22 makes a mistake in this weight information first Since register reference dependence with the VLIW instruction read from the instruction cache 22 just before it (carrying out an instruction-cache hit) cannot be inspected Since data forwarding is not made, it is for performing the VLIW instruction which inserts a wait cycle compulsorily, reads from a register file 27 even if it is as a result of [last] a VLIW instruction execution, and is executed next.
[0099] Next, operation at the time of the writing in an instruction cache 22 is explained.
[0100] In the instruction cache 22 shown in drawing 2 , the instruction address section and the LRU section consist of both content-retrievals type associative memories. At an instruction cache 22, existence of the block which is in agreement with the instruction address inputted the whole clock cycle is inspected in the state where it is not an instruction-cache mistake. This is performed by searching the instruction address section of each block of an instruction buffer. The instruction address section consists of content-retrieval type memory, and searches the block which holds the contents which are in agreement with the instruction address inputted from the outside in the instruction address section.
[0101] When a block in agreement exists, while negating a mistake signal, V flag of the block concerned, the number of instructions, and the contents of instruction 1 - n are outputted. Moreover, the LRU flag of the block concerned is simultaneously set to 1. A mistake signal is asserted when a block in agreement does not exist. In this case, n instruction codes read from instruction memory 21 one by one are stored in the same

block of an instruction buffer one by one.
[0102] At this time, the block which should newly store the instruction code for VLIW instruction generation is given from the tag portion of an instruction cache as a LRU block number. The LRU portion of an instruction cache 22 is content-retrieval type memory, and when writing in, it outputs the block number of the watch young No. 1 whose V flag is 0.
[0103] V flag of the block concerned is set to 0 at the same time a LRU flag outputs the block number of the watch young No. 1 which is 0, when all V flags are 1. Moreover, V flag of the block concerned is set to 0 at the same time it outputs the block of old age watch, i.e., m, most, when all V flags and all LRU flags are 1. Moreover, at the time of writing (at the time of renewal of a block), only the LRU flag of the updated block is set to 1, and the LRU flag of other the blocks of all is set to 0.
[0104] This method enables it to use the block in an instruction cache at the maximum efficiency.
[0105] In addition, the method of the LRU portion shown with the 1st operation form of operation of it not being limited to the method mentioned above, but your making it use the LRU mechanism by the existing technology is natural.
[0106] As explained above, the VLIW type parallel computer 20 concerning the 1st operation form It has instruction memory 21, an instruction cache 22, the instruction-cache write-in control section 23, the instruction address control section 24, two or more instruction-execution sections 25, a data switch 26, and a register file 27. From instruction memory 21, read 1 instruction code for every clock cycle, and it sets to the instruction-cache write-in control section 23. When register reference dependence with one or more VLIW instruction codes which were read immediately before and which consist of n instruction codes, respectively is inspected and reference dependence is detected While generating the number of wait cycles for inhibiting execution of the next VLIW instruction The switch control information for performing data forwarding is generated. the number of wait cycles, and switch control information It writes in the applicable block of an instruction cache 22 simultaneously with the read instruction code, and the VLIW instruction is beforehand constituted in the instruction cache 22. at the time of a VLIW instruction execution The VLIW instruction code which does not perform the register reference dependence check during a VLIW instruction, but only corresponds from an instruction cache 22, In order to refer to the VLIW instruction-execution result which read simultaneously the number of weight, and switch control information, and was performed immediately before, Since it constitutes so that VLIW instruction-execution queuing and data forwarding by the change of a data switch 26 may be performed Like the VLIW type parallel computer by the conventional example, at the time of a VLIW instruction execution It is not necessary to check the register reference dependence during a VLIW instruction, and to generate the control signal for suppression of the instruction execution by this, and data forwarding, and in case CPU is LSI-ized, it becomes possible to minimize reduction of a chip size, and circuit delay.
[0107] That is, in the VLIW type parallel computer by the conventional example, since it is necessary to each statement part to give a separate instruction to **********, VLIW instruction code becomes huge. For this reason, the buffer with the instruction memory 21 which installed the instruction cache in the interior of CPU, and was installed in the CPU exterior, and the instruction-execution section is

performed. For this reason, like the usual computer, instruction memory has many whose reference pack from an instruction cache to one VLIW instruction, and is enabled, though considered as the read-out width of face for one instruction. In such a VLIW computer, in order to minimize the queuing time of the instruction execution by the check and reference dependence of data reference dependence during each VLIW instruction, there were many things equipped with a data forwarding mechanism. However, the circuit scale and the amount of wiring for the combination of the reference dependence during a VLIW instruction checking very many (it being the order of N2, when ******** which carries out a concurrency is set to N) this become great. For this reason, when LSiota Changing CPU, problems, such as increase of a chip size and a fall of the clock frequency of operation by circuit delay, were caused.

[0108] on the other hand, in the VLIW type parallel computer 20 concerning this operation gestalt Like the VLIW type parallel computer by the conventional example, at the time of a VLIW instruction execution Since it is not necessary to check the register reference dependence during a VLIW instruction, and to generate the control signal for suppression of the instruction execution by this, and data forwarding The amount of circuits for checking the register reference dependence during a VLIW instruction serves as order of N also considering the throughput which carries out a concurrency as N. Therefore, the circuit scale and the amount of wiring for it serve as order of 1-/N of the VLIW type computer by the conventional technology.

[0109] In case CPU is LSI-ized, while becoming possible to minimize reduction of a chip size, and circuit delay, raising a clock frequency of operation and minimizing power consumption by this, there is an advantage which can reduce chip cost. Especially, Throughput N becomes remarkable [this effect], when large.

[0110] In the VLIW type parallel computer 20 concerning the 1st operation form mentioned above, although the instruction execution time was made into one clock cycle, the execution time of an instruction can apply also to two or more clock cycles or the parallel computer to cut. Hereafter, the 2nd operation form describes the example applied to the VLIW type parallel computer in case the execution time of an instruction cuts in a maximum of d clock cycles.

[0111] Drawing 6 is drawing showing the whole parallel computer composition concerning the 2nd operation form of this invention. The same sign is given to the same component as the VLIW type parallel computer shown in drawing 1 in explanation of the parallel computer concerning this operation form, and explanation of a duplication portion is omitted.

[0112] In drawing 6 , the VLIW type parallel computer 60 consists of instruction memory 21, an instruction cache 22, the instruction-cache write-in control section 61, the instruction address control section 24, a statement part 25 (statement-part 1-n), a data switch 26, and a register file 27.

[0113] It is getting down for the VLIW type parallel computer in case the execution time of an instruction cuts in a maximum of d clock cycles, and only this point is difference. Namely, it differs about the output signal from an instruction-cache control section in that that whose difference with the VLIW type parallel computer 20 shown in drawing 1 was a weight signal in the VLIW type parallel computer 20 serves as the number signal of weight in the VLIW type parallel computer 60 of this operation form. This is because the instruction execution time cuts in d clock cycle in the VLIW type parallel computer 60 of

this operation form. Except [all] this point, it is the same as the 1st operation form.
[0114] Drawing 7 is drawing showing the composition of the instruction cache of the VLIW type parallel computer 60, and the difference with the instruction cache shown in aforementioned drawing 2 is the point that what was a weight signal stores the number signal of weight in drawing 7 , in drawing 2 . Except [all] this point, it is the same as the 1st operation gestalt.
[0115] Moreover, the format of instruction code used as the element of the VLIW instruction executed by the VLIW type parallel computer 60 is the same as the 1st operation gestalt shown in aforementioned drawing 3 . Moreover, it is the same as the operation gestalt 1 which also shows the composition of the instruction address control section 24 in the VLIW type parallel computer 60 to aforementioned drawing 5 .
[0116] Drawing 8 is drawing showing the composition of the instruction-cache write-in control section 61 of the above-mentioned VLIW type parallel computer 60, gives the same sign to the same component as the instruction-cache write-in control section 23 of the VLIW type parallel computer 20 shown in drawing 4 , and omits explanation of a duplication portion.
[0117] In drawing 8 , the instruction-cache write-in control section 61 consists of an instruction register 31, a shift register 32 (shift register 1-n), latch 62, a comparator 63, a buffer 64, OR circuit 65, a selector 66 (selector 1), a selector 67 (selector 2), an instruction counter 36, a decoder 37, AND circuit 38 (AND circuit 1), AND circuit 39 (AND circuit 2), OR circuit 40 (OR circuit 1), and D latch 41.
[0118] The instruction-cache write-in control section 23 shown in aforementioned drawing 4 and the following points are different. Namely, latches 62 are 11 to d-1n of latches, and x(d-1) n piece, A comparator 63 is 111 to d-1n [of comparators] 2 and x(d-1) nx2 pieces, d-1 OR circuit 65 to 2 and Orr d who takes d-111 to d-1n of all ORs of the output of 2 was added [that a buffer 64 is 111 to d-1n / of buffers / 2 and x(d-1) nx2 pieces, and] 111-1n 2,211-2n comparator -- it comes out
[0119] The output of these d-1 OR circuits 65 serves as way ** d-1, weight d-2, --, weight 1 signal, respectively.
[0120] Moreover, flip-flop 42 output serves as a weight d signal. furthermore, buffers 111 and 121, --, the output of 1n1 -- a wye yard -- or it carries out -- having -- buffers 211 and 221, --, the output of 2n1 -- a wye yard -- or it carries out -- having -- the following -- the same -- a buffer d-111, d-121, --, the output of d-1n1 -- a wye yard -- or it is carried out and these d-1 signal is inputted into a selector 1
[0121] moreover, buffers 112 and 122, --, and the output of 1n2 -- a wye yard -- or it carries out -- having -- buffers 212 and 222, --, and the output of 2n2 -- a wye yard -- or it carries out -- having -- the following -- the same -- a buffer d-112, d-122, --, and the output of d-1n2 -- a wye yard -- or it is carried out and these d-1 signal is inputted into a selector 67
[0122] A selector 66 (selector 1) and a selector 67 (selector 2) It is the circuit which chooses an output with d-1 weight signal to weight d-1-1. The wye yard or the buffer output carried out corresponding to the biggest weight signal of the number of weight currently asserted among these weight signals is chosen. Respectively, considering-as data forward 1 signal and data forward 2 signal ** is the difference with the instruction-cache write-in control section 23 of the operation form of the above 1st.
[0123] All of these differences are in the VLIW type parallel computer 60 of this

operation form for generating the number signal of weight for inhibiting the following VLIW instruction execution a maximum of d clock cycles for queuing of that the instruction execution time needs to be based on a maximum of d clock cycles and a bird clapper, and it is necessary to perform the register reference dependence check during a VLIW instruction about between d VLIW instructions generated continuously and, and refer to the data. It is the same as the instruction-cache write-in control section 23 of the operation form of the above 1st, and the same is [all of except for these differences] said of the function and the operation.

[0124] Hereafter, operation of the VLIW type parallel computer 60 constituted as mentioned above is explained.

[0125] In the VLIW type parallel computer 60 of this operation form, operation in an instruction-cache hit is almost the same as the VLIW type parallel computer 20 of the operation form of the above 1st. A different point is that the weight information read from an instruction cache 22 is not weight specification of one mere clock cycle but the number of wait cycles to 1-d. By the VLIW type parallel computer 60 of this operation form, the instruction execution time depends this on a maximum of d clock cycles or cutting. Therefore, each statement part will wait the clock cycle time instruction execution corresponding to the number of weight inputted.

[0126] It is completely the same as the case of the 1st operation form at the point of reading instruction code from instruction memory 21 one by one in an instruction-cache mistake, and generating the VLIW instruction which was a mistake being made in with writing one by one at the applicable block of an instruction cache 22. It differs that the check of the register reference dependence during a VLIW instruction is performed to each instruction code used as the element of the VLIW instruction which is newly going to be generated among d-1 VLIW instruction generated in the instruction cache 22 just before it so that the composition of the instruction-cache write-in control section 61 of drawing 8 may see.

[0127] In order that an instruction execution may take this a maximum of d clock cycles, it is needed for the generation of switch control information for the data FO wording immediately after the number information generation of weight required for instruction-execution suppression and the resumption of an instruction for reference queuing to the execution result of d-1 last VLIW instruction.

[0128] In drawing 8 , it is the same as that of the case of the operation form of the above 1st that it is the changes detector 43 which generates the signal with which the Arthur ** only of the first one clock cycle which the not continuous instruction-cache mistake generated is carried out of the D latch 41, OR circuit 40 (OR circuit 1), AND circuit 39 (AND circuit 2), and a flip-flop 42.

[0129] However, the weight signal generated by this changes detector 43 Also in a maximum of d clock cycles or which instruction code of d-1 VLIW instruction executed just before cutting Though the instruction code which is the element of the VLIW instruction which it is newly going to generate in an instruction cache 22 refers to the register made applicable to updating as a destination operand as a source operand, it After a register file 27 is surely updated, in order to guarantee reading the data from a register file 27, and referring to it, it becomes a weight signal for inhibiting the instruction execution of d clock cycle.

[0130] Like the operation form of the above 1st, when an instruction cache 22 makes a

mistake in this first Since register reference dependence with the VLIW instruction read from the instruction cache 22 just before it (carrying out an instruction-cache hit) cannot be inspected Since data forwarding is not made, it is for performing the VLIW instruction which inserts a wait cycle compulsorily, reads from a register file 27 even if it is as a result of [last] a VLIW instruction execution, and is executed next. A difference is based on the point which is the a maximum of d clock cycle need at a VLIW instruction execution. Switch control information is generated so that data forwarding may surely work to required destination operand data the maximum queuing time.

[0131] That is, though two or more weight specification signals, i.e., the plurality of weight d-1-1, are simultaneously asserted in OR circuit 2 - d, all the signals of weight d-1-1 are written in the applicable block of an instruction cache 22 as number information of weight. At an instruction cache 22, in order to supply this information to a statement part as it is, though the plurality of weight d-1-1 is asserted, by the statement part, the instruction execution of a clock cycle equal to the number of the maximum weight of them is inhibited as a result. At this time, in a selector 1 and a selector 2, weight specification chooses and outputs the greatest destination operand, and writes the contents of each switch control information of the data forward 1 and the data forward 2 in the applicable block of an instruction cache 22 as switch control information, respectively. When the number of weight is d clock cycle, data forwarding is not performed, but surely read and refer to the source operand data for it from a register file 27.

[0132] The difference with the instruction-cache write-in control section 23 of the operation gestalt of the above 1st In the VLIW type parallel computer 60 of this operation gestalt The instruction execution time needs to be based on a maximum of d clock cycles and a bird clapper, and it is necessary to perform the register reference dependence check during a VLIW instruction about between d VLIW instructions generated continuously, And it is for generating the number signal of weight for inhibiting the following VLIW instruction execution a maximum of d clock cycles for queuing of refer to the data. It is the same as the instruction-cache write-in control section of the operation gestalt 1, and the same is [all of except for these differences] said of the function and the operation.

[0133] As explained above, the VLIW type parallel computer 60 concerning the 2nd operation form While the instruction execution time considers as a maximum of d clock cycles and performs the register reference dependence check during a VLIW instruction about between d VLIW instructions generated continuously Since the function which generates the number signal of weight for inhibiting the following VLIW instruction execution a maximum of d clock cycles is had and constituted for queuing of refer to the data, in addition to the ability to acquire the same effect as the 1st operation form, the following effects can be acquired.

[0134] That is, when the throughput of one VLIW instruction is set to N, in the case of the VLIW type parallel computer by the conventional example, a circuit scale required for the register reference dependence check during a VLIW instruction serves as order of N2xd, and becomes with the order of Nxd by the case of the VLIW type parallel computer 60 of the 2nd operation form. This circuit scale of 1-/N and a bird clapper is the same as that of the case of the 1st operation form. Especially, with this operation form, the instruction execution time can respond to the VLIW type parallel computer of a maximum of d clock cycles, and since it is more general, compared with the VLIW type

parallel computer 20 of the 1st operation form, a scope is in a large thing.
[0135] Thus, the VLIW type parallel computers 20 and 60 concerning the operation form concerning the 1st and 2nd operation form In the VLIW type parallel computer which performs simultaneously processing which became independent, respectively in two or more instruction-execution sections Between the instruction-execution sections 25 with the instruction memory 21 installed in the CPU exterior While installing an instruction cache 22, reading two or more VLIW element instructions into an instruction cache 22 serially and generating a VLIW instruction continuously from instruction memory 22 by this Register reference dependence with the VLIW instruction generated immediately before is inspected. Required queuing information, And since it is not necessary to detect or extract these at the time of an instruction execution as it has the mechanism in which the information about a setup of the path for the bypass of execution result data is extracted beforehand and carried out by the VLIW type parallel computer by the conventional example, Curtailment of the amount of circuits and evasion of a fall of the CPU working speed by great circuit delay are possible.
[0136] In addition, with each above-mentioned operation form, although the example of adaptation to a VLIW type parallel computer was shown, if it is the parallel computer which is not what especially an instruction length gives a limit, either, and performs simultaneously processing which became independent in two or more instruction-execution sections, respectively, it can be adapted. Moreover, the above-mentioned instruction feeder cannot be overemphasized by that you may be a part of circuit included in a computer etc.
[0137] Moreover, numbers, such as a buffer which constitutes each above-mentioned control section etc., a register, a logical circuit, and a comparator, a kind connection state, etc. cannot be overemphasized by not being restricted to the above-mentioned operation form.
[0138]
[Effect of the Invention] In the parallel computer concerning this invention, instruction memory, an instruction cache, an instruction-cache write-in control section, When the VLIW instruction which should be equipped with an instruction address control section, two or more instruction-execution sections, a data switch, and a register file, and should next be executed does not exist in an instruction cache Every one instruction code used as the element of a VLIW instruction is continuously read from instruction memory for every clock cycle. It sets to an instruction-cache write-in control section at the same time it writes in the block made into the write-in object in an instruction cache serially. Consist of instruction code and instruction code of plurality respectively which read into just before from instruction memory, and was generated. When the register reference dependence between one or more VLIW instruction codes is inspected and reference dependence is not detected The value as which it was beforehand determined for supplying the read-out data from a register file to the instruction-execution section as data switch control information When it writes in the applicable block of an instruction cache and reference dependence is detected The number of wait cycles for inhibiting execution of the next VLIW instruction is extracted. A data switch is minded for the execution result data of the VLIW instruction executed immediately before from the instruction-execution section without a register file at the same time it writes this number of wait cycles in the applicable block of an instruction cache. The data switch control

information for supplying the specific instruction-execution section of two or more instruction-execution sections is extracted directly. This data switch control information is written in the applicable block of an instruction cache. Then, the newly generated applicable VLIW instruction is read in an instruction cache. The read VLIW instruction All the instruction-execution sections, a data switch, and register stuffer IRUHE, When the VLIW instruction which should supply simultaneously, should execute this VLIW instruction and should next be executed exists in an instruction cache Since it constitutes so that the VLIW instruction which should be this executed may be read from the applicable block of an instruction cache, all the instruction-execution sections, a data switch and register stuffer IRUHE, and **** may be supplied and this VLIW instruction may be executed A circuit scale and the amount of wiring are sharply reducible, and in case CPU is LSI-ized, reduction of a chip size and circuit delay can be minimized. Consequently, chip cost can be reduced, while raising a clock frequency of operation and minimizing power consumption.

[Translation done.]